Session A6: Implications of M&S Foundations

Session A6 leaders:

Co-Chairs:

Michael Moulding (Cranfield University, UK)
Dave Thomen (Army Modeling and Simulation Management Office)

A6 Materials in Foundations '02 proceedings:

<u>Paper</u>

Implications of M&S Foundations for the V&V of Large Scale Complex Simulation Models (50 pp)

Bernard P. Zeigler (University of Arizona)
Hessam S. Sarjoughian (Arizona State)

<u>Slides</u> (may contain back-up materials and notes)

Implications of M&S Foundations for the V&V of Large Scale Complex Simulation Models (78 slides) [A6B in both pdf and ppt formats]

Bernard P. Zeigler (University of Arizona)
Hessam S. Sarjoughian (Arizona State)

Participants in this session are listed at the end of the Discussion Synopsis.

Discussion Synopsis (to provide perspective on papers & briefings identified above).

A summary of discussion issues during A6. (Content of basic presentation and paper is not summarized).

Since the Discrete Event System Specification (DEVS) is unfamiliar to most in the audience, the primary presenter, Zeigler, devoted the first half of the session to a tutorial on DEVS. In the process, he indicated that the technology has been implemented in industrial strength environments and is currently being applied to system-of-systems formulation of an overall architecture for all C4ISR projects within a major defense contractor.

A significant portion of the discussions revolved around the applicability to continuous systems such as physics problems – ordinary and partial differentioal equations (PDEs). An attendees raised the issue in terms of tightly coupled physics problems, using the example of turbulent combustion. Separately, turbulence and combustion are difficult enough to simulate using PDEs, but present additional complications when coupled. Zeigler demonstrated DEVS on a relatively simple PDE model - diffusion - and claimed that more complex examples do not introduce additional challenges to the fundamental methodology. The key is to formulate the individual components with the appropriate input and output ports through which all interactions are made explicit.

Another attendee inquired about the capability to track numerical error propagation (and similar issues in computational mechanics). Zeigler replied that the basic representation of events within continuous models was determined by a quantum (minimum threshold difference for sending an output or state update to other components). A fundamental representation theorem shows that given any desired accuracy for the DEVS model's overall behavior, there was a choice of quantum size small enough to achieve this accuracy. The trade-off between speed and accuracy is controlled by this choice. However, the attractive feature of DEVS representation is that there is an significant inherent computational advantage in discrete event simulation both in terms of number of computations on a single machine and reduction in bandwidth requirements in distributed simulation.

A question concerning the capability of DEVS to support successive refinements of components was raised. Zeigler replied that DEVS hierarchical modular construction accommodates dynamic structure changes (while the simulation is in progress) such as replacing an atomic component by a coupled model of many components. Zeigler also raised the relation of dynamic structure in DEVS to adaptive mesh refinement (AMR) a state-of-the-art technique in conventional PDE solution. The comparison between the two is a question of current research that can be approached directly by replicating the dynamic mesh refinement within DEVS or perhaps by an equivalent non-dynamic approach. In the latter, one would use the finest resolution that was employed in a run of AMR as the fixed resolution for DEVS. Since the DEVS method inherently focuses its calculations in regions according to their activity, the regions where AMR refinement occurs will be those in which it concentrates its attention. Other regions will be interrogated and ignored if inactive (of course memory has to be big enough or dynamic memory has to be used to pull active cells into working memory when needed). Another attendee noted that current computational methods in distributed simulation can incorporate such active-region-only logic. Zeigler noted that the extra feature was that this arose naturally as a property of the DEVS representation of continuous systems for distributed simulation. He remarked that in fact the original impetus for the DEVS representation of such problems was a DARPA funded effort to formulate the underlying theoretical justification for the predictive contract (dead reckoning) techniques that were used in DIS (distributed interactive simulation) to mitigate network latencies and bandwidth limitations.

Co-Author Sarjoughian's focus during second half of session was on V&V implications of DEVS. By separating the "model" [the "conceptual model" in DoD parlance] from the "simulator" [the implementation of the model], he claimed that a much better handle on V&V for large scale distributed simulations can be had. The software related aspects of V&V are concerned with the correctness of the simulator in executing the transition rules specified by the model. This involves verification as normally considered by practitioners of formal methods for software development. Once done for a class of models, such as all DEVS models, such verification need not be continually revisited as it must when both model and simulation are intertwined in program code. Clarification on "simulator" was requested. Is "simulator" a virtual machine from which multiple simulations can be created rather than a single purpose program with a restricted functionality? Sarjoughian replied that a "simulator" is conceptually a device for executing model instructions and therefore can be designed to be correct for a class of models of any size ranging from one to an indefinite number depending of the objectives of the development. The question was raised, does this make DEVS universal in some sense. Zeigler replied that a theorem was proved that showed that indeed DEVS was universal in the sense of being able to represent the models defined by any formalism for discrete event systems, such as Petri nets or process algebras. This DEVS universality depends on the assumed universality of the underlying mathematical systems theory to express any dynamic time-based system. He noted that theorems such as these are found in the second edition of *Theory of Modeling and Simulation*.

It was noted that DEVS allows opening up the M&S process to a new way of thinking, including a way to help formalize the requirements.

Sarjoughian detailed the multiple levels/components of the model development and application process, stressing that assessment at each level/component brings an important aspect to the overall assessment. The separate assessments include usability, validity, etc. Some noted the opportunity to "measure where you are", likening it to the objective of others to create validation metrics.

Zeigler noted that a major necessity, and therefore objective of the further development of the theory and framework for modeling and simulation, is to create a more generally understandable form of the formalism and support it with tools for training people to grasp the underlying concepts. There was an increasing need for such training of existing modeling and simulation personnel who have found themselves in these functions without having the requisite background. It was also very much needed to support new curricula for college and university degree tracks in M&S.

The following points were also noted during the session:

A deficiency with M&S V&V today is that there is no way to intrinsically check that a model fits the purpose for which it is to be used.

A foundation of underlying theory is needed before we can really begin to address questions of verification and validation. Among other things, people will not share a common terminology, and will have different views on what is being done. That is why well-founded concepts and a mathematical underpinning is essential. Such formal methods must exist so you know where the verification and validation methods fit in.

It was observed that what was being presented was a type of formalism for looking at a problem. However, whereas traditional logical formalisms attempt to understand the problem and analyze the properties of the problem in order to develop an approach for analyzing the problem itself, this formalism is focused on how to present the problem in a way that you can simulate it. Specifically, it permits you to “animate” the problem. The authors concurred with this observation.

There was some clarifying discussion over the statement that often in code you have both elements of the model and elements of the simulator. Therefore, it is important to separate the two, since verification looks at the relationship between the model and the simulator; validation is focused on the relationship between the output of the simulator as it executes the model and the source system in its experimental frame.

There was clarifying discussion on the difference between a “base model” and a lumped model (reference slide 39 in the authors’ presentation). It was stated that the base model is a conceptual model, and is the most extensive model that contains all the entities needed to answer all the questions as formulated in an experimental frame. The lumped model is consistent with the conceptual base model but is limited in that it can only answer a portion of the questions. For example, it may only be able to address only part of the available data as required by some experiment frame. The conceptual base model “refines” the lumped model. An experimental frame helps the lumped model developer to focus in on what is needed to answer the questions, and to omit the unneeded. This is a process of “coarsening” and omitting detail that is not relevant to the objective of the project. It contrasts with the reverse process of “refinement” in which detail is added to the model. As you get closer and closer to the conceptual base model, your lumped model is refined with the possibility of addressing more and more experimental frames – but since the added complexity can be very costly in terms of time and money, it is advisable to specify the minimum experimental frame required for the current objectives and to only develop a lumped model for it. Note: the modeler views the “base vs lumped” in a different perspective than, say, a software coder. For the latter, the task is to take the lumped model which is abstractly specified and to refine it successively toward more and more implementation details until finally a working program code is obtained.

The authors state that the FEDEP provides the generic concept, but it does not provide details of how to execute it. It offers generalized guidelines on how federation development steps correspond to the steps of the VV&A process. It does not tell us the functionality it is suppose to have and the requisite relationships among the steps and objects (see slide 40 in the authors’ presentation). For example, if what you develop in step 3 is not homomorphic with step 2, the utility of the federation is compromised. What is needed is a mechanism that allows the federation developer to tie the steps together. The framework for M&S and the DEVS formalism provide such a mechanism.

A6 Session Participants (14)

Geoffrey	Hone	Cranfield University
Hans	Mair	Institute for Defense Analyses
Michael	Moulding	Cranfield University
William	Oberkampf	Sandia National Laboratories
Bill	Ormsby	Naval Surface Warfare Center
Sharon	Padula	NASA Langley
Steve	Painter	Los Alamos National Laboratory
Paul	Pleva	Jacobs Sverdrup
Hessam	Sarjoughian	Arizona State University
Randall	Shumaker	Institute for Simulation and Training
Bill	Stevens	NASA Langley
David	Thomen	Army Model & Simulation Office
V Gregory	Weirs	The University of Chicago
Bernard	Zeigler	AZ Ctr for Integrative M&S